ARMY RESEARCH LABORATORY

Web-Based Mesoscale Model Execution and Evaluation Tool: A Prototype

by Stephen F. Kirby

ARL-MR-548 January 2003

Notices

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

Citation of manufacturers' or trade names does not constitute an official endorsement or approval of the use thereof.

Army Research Laboratory
White Sands Missile Range, NM 88002-5513

ARL-MR-548 January 2003

Web-Based Mesoscale Model Execution and Evaluation Tool: A Prototype

Stephen F. Kirby
Computational and Information Sciences Directorate, ARL

Contents

Figures

Summary

U.S. Army personnel are highly dependent on receiving the highest quality meteorological data possible, either for missions or for training. This data may be input to a tactical decision aid ((TDA) a software component intended to assist military personnel in formulating plans), the basis on which troop movements are made, a determining factor on how armaments are employed, etc. Thus, when the required data is derived from models, the models must be thoroughly evaluated for the accuracy of the output prior to integration into training scenarios and the battlefield.

To provide a more streamlined method by which U. S. Army Research Laboratory researchers can both execute and evaluate mesoscale models, a prototype Java Server Pages-based Web site has been developed. Through this site, researchers can:

1. Execute the Fifth Generation Mesoscale Model (MM5) (other models are to be added).
2. Download "truth" data, i.e., actual field measurements. In this case, mesonet meteorological data from the Mesowest mesonet, which encompasses much of the west and southwestern United States.
3. Generate correlation coefficients based on the MM5 output and the acquired truth data (other statistical measurements are to be added).
4. Create Java-based visualization toolkit (VisAD-based) graphics for viewing the correlation coefficient values.
5. Store the VisAD graphic Joint Photographic Experts Group (JPEG) values in a Postgresql database table.

This report examines the tools behind this Web site, which includes Java Server Pages, VisAD and Postgresql database software. Also discussed are the details behind bilinear interpolation, which is used to create meteorological values from MM5 that are co-located with the truth data. Sample results are given showing temperature correlation coefficients generated for a 24-h MM5 forecast. Lastly, ways to enhance the capability of this system are considered.

Introduction

Researchers at the U.S. Army Research Laboratory (ARL) investigate the accuracy of mesoscale meteorological models to determine which models are best suited for U.S. Army applications and requirements. The applications and requirements range from providing meteorological input data for tactical decision aids, to the display of meteorological parameters, such as wind or temperature fields, for a Staff Weather Officer. To facilitate a means for both automating and simplifying this model evaluation process, a prototype Model Execution and Evaluation Tool (MEET) has been developed.

The model being tested in this prototype is the 5th Generation Mesoscale Model (MM5), a meteorological model developed jointly by Pennsylvania State University and the National Center for Atmospheric Research. Navy Operational Global Atmospheric Prediction System (NOGAPS) data provides the large-scale initialization for MM5 and is at 1° horizontal resolution (i.e., about 111 km). NOGAPS provides MM5 with fields of winds, temperatures, moisture values, and geopotential height at pressure levels, at 6-h intervals beginning at 00:00 UT (universal time) through 24:00 UT. These are the values toward which the MM5 forecast fields are steered.

The statistical method employed for this prototype is the Pearson correlation coefficient. The formula for the Pearson correlation coefficient used in MEET, is given by:

$$\left(\sum_{i=1}^{n} x_i y_i - (1/n)\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)\right) / \left(\left[\sum_{i=1}^{n} x_i^2 - (1/n)\left(\sum_{i=1}^{n} x_i\right)^2\right]^{1/2}\left[\sum_{i=1}^{n} y_i^2 - (1/n)\left(\sum_{i=1}^{n} y_i\right)^2\right]^{1/2}\right) \quad (1)$$

where

n represents the number of measurements of some meteorological parameter; x is the MM5 value interpolated to some mesonet[1] station location, and y is the mesonet station value. (2).

This report will provide background information for both the software components of MEET and the methodology behind MEET, describe the MEET user interface and options, then give sample statistical results using MM5. Lastly, future plans to enhance MEET will be discussed.

1. MEET Software Components

The Model Execution and Evaluation Tool has been built almost entirely using open-source[2], freely available software, including the operating system, Linux, the mesoscale model, MM5, and the database software, Postgresql. Though they are open source, all of the software entities used to build MEET are of very high quality, because each software component has a very large user base, which implies exhaustive testing (by virtue of the number of users) and constant improvement. Each of these components are also maintained by a central controlling body. This group maintains the software releases and

[1] A mesonet is an array of automated surface meteorological sensors (1).

[2] The only exceptions are the Portland Group compilers, which are required for MM5 compilation.

will make software improvements and/or bug fixes based on their findings or the findings of some external person, often through forums on the Internet. By avoiding vendor-specific software packages, greater flexibility has been built into MEET, in that software changes to the source code can be made at any point. Also, updating the various software packages is streamlined since vendor dependence, in the case of the open-source software, is eliminated. A description of each of the open-source software components within MEET is given below:

Java Server Pages (JSP). JSP files serve as the backbone for this effort. A JSP file can contain:

1. Template data: Template data refers to the static portions of Hypertext Markup Language (HTML), eXtensible HTML (XHTML) and text.

2. JSP directives: JSP directives control the translation and compilation process and do not produce any output. They may perform such tasks as importing a Java class utility or setting an attribute value.

3. Dynamic data: By using dynamic data, the programmer can insert code that is translated and then, generally, displayed. A user could, for example, import the date utility by using a JSP directive and then generate a date value using a dynamic data call, which is subsequently displayed using HTML, XHTML, and so on.

JSP allows a programmer to embed Java code directly within HTML, XHTML, etc. The embedded Java code performs such functions as gathering client information, and performing an action based on those inputs. For this project, the JSP calls XHTML, which requests information from the user regarding what model forecast times are to be used, how many nested domains are required, which nested domain to generate statistics for, etc. JSP is preferable to Common Gateway Interface (CGI), another popular method for presenting Web pages, because every time a request is made for dynamic content, a CGI program must be executed. Thus, if a number of users enter a Web site and request information, a CGI program is launched for each user. As the number of users increases, the competition for CPU (Central Processing Unit) time grows. If Perl is the CGI program language, the problem is exacerbated because Perl code is interpreted (2).

Web server. The Web server used for this study is a Tomcat Web server. A user typically develops their JSP Web application underneath the "webapps" directory within the Tomcat software tree. Once a JSP application is built, the tree containing its components can be combined into one file, called a Web archive (*.war) file, which is created by archiving all of the user's code underneath the webapps directory. This provides great flexibility because the *.war file can be brought onto any other computer running a similarly configured Tomcat server, and then expanded. The newly configured computer is immediately ready to run the JSP application.

XHTML. XHTML is the markup language used for this project, and is compatible with eXstensible Markup Language (XML) (i.e., conforms fully to XML standards). Because XML is the markup language that provides the most flexibility to the programmer, the ultimate goal is to use strictly XML as the markup language for MEET. However, significant development is required, for example, document type definitions (DTD) or grammar rules for parsing the XML must be written. As a first step toward moving beyond HTML to XML, XHTML is being used, thus establishing a familiarity with XML grammar.

VisAD. VisAD, developed at the University of Wisconsin, is a toolbox of Java-based visualization routines for the 2-d or 3-d visualization of meteorological parameters. VisAD allows a user to display such things as line graphs, isosurfaces and vertical slices. For this prototype, line graphs are used to illustrate the MM5 error. However, VisAD's ability to display other means of rendering statistical analysis, such as scatterplots, is intended to be incorporated in the next revision of MEET.

Postgresql. Postgresql is the relational database software chosen for this project. Postgresql is freely available and has a large user base. Thus, it has been, and continues to be, thoroughly tested and improved.

Java Spaces. Java Spaces is a software paradigm that makes distributed programming much simpler. This is done through the creation of a "virtual space," or, Java Space. Routines called "producers" create tasks and write them to the virtual space. Routines called "workers," also called processors, monitor the Java Space for new tasks and then complete them. When the worker finishes its task, it will write the result back to the Java Space. Running parallel to these two entities are "collectors," which monitor the Java Space and look for new results. When they find the number of results they expect to see, they consider the job completed and typically write out the final results (3). As part of this project, Java code, which utilizes Java Spaces specifically for generating correlation coefficients, has been written but is not yet integrated into the JSP code (4).

2. MEET User Interface and Option Description

The MEET user interface (UI) is generated by XHTML as directed by JSP. The UI for MEET (shown in figure 1) is the point where the JSP program gathers all key user inputs. The prototype version of MEET, described in this report, requires that the user enter the minimum information necessary to start a MM5 run. Future enhancements will allow the user to set various "physics" options, such as what cumulus parameterization type to be used. Also, note that the only mesoscale model currently available is MM5. The Weather Research and Forecasting (WRF) model is to be added, as denoted by the fields labeled as placeholders in the current UI.

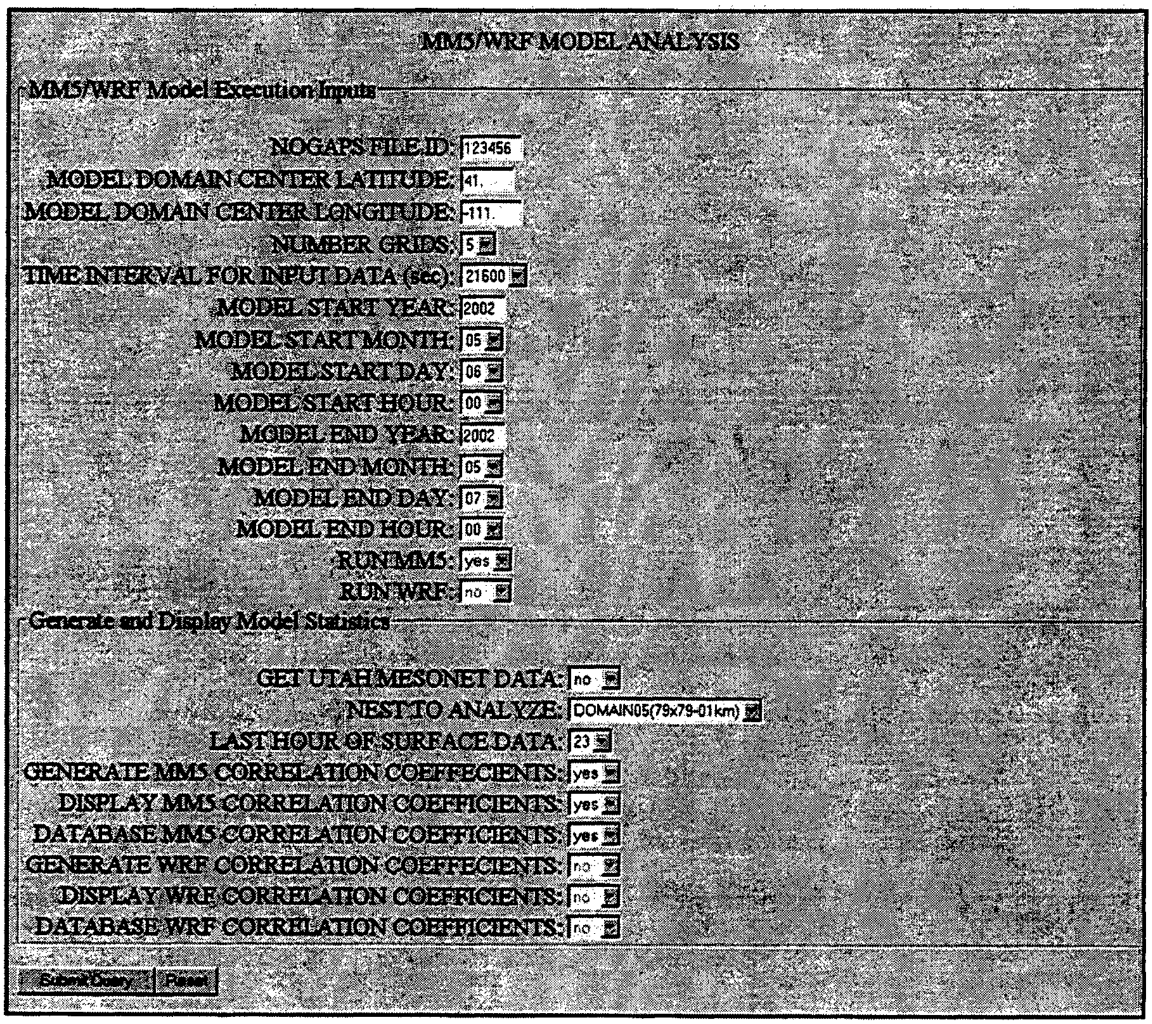

Figure 1. MEET UI.

The following are descriptions of the UI fields shown in figure 1:

1. NOGAPS FILE ID: The user enters the number string given to the file by NOGAPS. NOGAPS is the large-scale data used to initialize the MM5 model run, and is available through the Master Environment Library Web site. Each file is given a name composed of static characters plus a number string.

2. MODEL DOMAIN CENTER LATITUDE: This value, in conjunction with longitude, will serve as the center point for the "coarse" domain, or what is typically the outermost nest.

3. MODEL DOMAIN CENTER LONGITUDE: This value, in conjunction with latitude, will serve as the center point for the "coarse" domain, or what is typically the outermost nest.

4. NUMBER GRIDS: The user specifies up to five nested domains. The coarse domain is set at 81 km grid spacing. The next four domains are set at 27, 9, 3, and 1 km grid spacing.

   Regardless of the user's choice, a recompilation of MM5 will occur which is transparent to the user. Note that since the model is currently hosted on a computer with a 1 GHz processor, the combination of the automatic recompilation and the 3 km and 1 km resolution domains increases run times substantially. Increasing grid resolution to 1 km on the 1 GHz system can be prohibitively expensive (for example, a 24-h forecast using 5 nested domains and an innermost nest resolution of 1 km requires more than 72 hours).

5. TIME INTERVAL FOR INPUT DATA (sec): The user enters the interval between each NOGAPS data file. For the cases in this report, 21,600 s (6-h) is used.
6. MODEL START YEAR: The user chooses the model start year.
7. MODEL START MONTH: The user chooses the model start month (1–12).
8. MODEL START DAY: The user chooses the model start day (1–31).
9. MODEL START HOUR: The user chooses the model start hour (00–23).
10. MODEL END YEAR: The user chooses the model end year.
11. MODEL END MONTH: The user chooses the model end month.
12. MODEL END DAY: The user chooses the model end day.
13. MODEL END HOUR: The user chooses the model end hour.
14. RUN MM5: The user chooses "yes" or "no."
15. RUN WRF: This is currently a placeholder entry and is disabled.
16. GET UTAH MESONET DATA: The user chooses "yes" or "no."

    Selecting yes provides the latest available 24-h set of surface observations for the southwestern United States (Mesowest[3]). The observations will be sent via file transfer protocol (FTP) to the server.

17. NEST TO ANALYZE: The user selects one of the grid size options below, and the correlation coefficients will be generated:

    DOMAIN01(41x41-81km)

    DOMAIN02(52x52-27km)

    DOMAIN03(70x70-09km)

    DOMAIN04(76x76-03km)

[3] The Mesowest mesonet encompasses much of the west/southwestern U.S. (1).

DOMAIN05(79x79-01km)

18. LAST HOUR OF SURFACE DATA: The user enters the last hour of the surface reports that the user has downloaded from the University of Utah Mesowest mesonet data FTP site.

19. GENERATE MM5 CORRELATION COEFFICIENTS: The user chooses "yes" or "no."

    If yes is selected, Pearson correlation coefficients are generated for these parameters: temperature, relative humidity (RH), and the u- and v-components of the wind.

    Note that RH is not provided by MM5; instead, it must be derived from mixing ratio, temperature, and pressure values.

20. DISPLAY MM5 CORRELATION COEFFICIENTS: The user chooses "yes" or "no."

    If the user selects yes to display the MM5 correlation coefficients, Java code is executed, which employs VisAD classes to render a line graph. One of the VisAD classes used saves the VisAD display to a JPEG file.

21. DATABASE MM5 CORRELATION COEFFICIENTS: The user chooses "yes" or "no."

    Currently, it is this JPEG file that is saved in the Postgresql database and is time-stamped with the MM5 model forecast start time.

22. GENERATE WRF CORRELATION COEFFICIENTS: This is currently a placeholder entry and is disabled.

23. DISPLAY WRF CORRELATION COEFFICIENTS: This is currently a placeholder entry and is disabled.

24. DATABASE WRF CORRELATION COEFFICIENTS: This is currently a placeholder entry and is disabled.

Once the user has provided the required information, MEET is prepared to do any or all of the following:

1. Execute MM5.
2. Obtain "truth" data from the Mesowest mesonet data FTP site.
3. Generate correlation coefficients.
4. Display correlation coefficient.
5. Place JPEG data from VisAD rendering of correlation coefficient into Postgresql table.

The following section provides the details behind each of these steps.

3. MEET Data Processing Flow and Sample Output

After the user has made the required entries, the JSP program then saves each value in local string variables. Should the user wish to run MM5, a terrain "namelist" file is generated based on the user inputs. The namelist file is used by a FORTRAN executable to generate up to five terrain files, which contain the elevation and land-use (vegetative cover, etc.) data interpolated onto grids. The Java Runtime class enables the JSP program to call a shell script, which in turn runs the FORTRAN program.

Next, the JSP sets up symbolic links so that files created under the PREGRID directory point to the actual files in the NOGAPS repository. Again, a namelist file is created based on user inputs. The namelist file provides the required inputs to the FORTRAN executable. The Java Runtime class calls the PREGRID executable. Variables that are outlined in a file called Vtable.NOGAPS are extracted by the PREGRID program. A Vtable.* file is where the gridded binary parameter IDs are associated with a parameter name. If there are gaps in the file times, PREGRID will interpolate. The REGRID program then uses the horizontal fields from PREGRID and interpolates the data horizontally onto the MM5 model grid, making it conform to the terrain grids.

The last preprocessing program is INTERPF, which is called by the Java Runtime class after the original namelist file is created. The purpose of the INTERPF program is to interpolate the data onto sigma levels, and to establish initial conditions and boundary conditions for the coarsest grid. The sigma coordinate used in MM5 is given as:

$$\sigma = (p - p_{top}) / (p_{s0} - p_{top}) \tag{2}$$

where

p is pressure, p_{top} is pressure used at model top, p_{s0} is the surface pressure.

For this prototype, $p_{top} = 50$ mb and $p_{s0} = 1000$ mb. Thus, $\sigma = 0$ at the model top and 1 at the surface.

At this point in the process, the majority of the input files needed to run MM5 are ready, and MM5 must be configured for the grids that have been prepared. Before the MM5 program is compiled, the JSP generates a configuration file based on user input. It is this configuration file that determines the array sizes, physics options, etc., which will be built into the MM5 executable. The Java Runtime class is again employed through the JSP to initiate the MM5 compilation. The final step, prior to starting the MM5 run, is to generate the input file, "mmlif," which specifies parameters, such as the forecast length, time step, file output frequency, and physics options.

Lastly, the JSP triggers the MM5 run. On the current Linux server, which is a single 1 GHz processor system, a 3 domain, 24-h forecast benchmark (45x45 – 81 km resolution, 61x61 – 27 km resolution, 115x115 – 9 km resolution) took approximately 6 hours.

If the user chooses not to execute MM5, the JSP then determines whether any of the following options have been selected:

1. If the user has opted for obtaining the University of Utah Mesowest mesonet data, an FTP session is begun to capture the most recent 24-h set of surface station reports.
2. If the user elects to generate MM5 statistics, FORTRAN and C code, used to complete this task, is generated, compiled, and run through the JSP.
3. If the user enables the option to view the statistics, the JSP initiates Java code, which in turn uses VisAD classes to generate a line graph of the values, and create a JPEG file of the image.
4. If the user chooses to place the JPEG image data in a Postgresql database table, the JSP will search for a JPEG file. If one is found, it is placed in the Postgresql database, assigned an identifying key by Postgresql, and time-stamped with the initial forecast hour run-time.

After the user options are processed, the JSP completes its task by displaying the time stamp, the identifying key of the JPEG in the database table, and the VisAD-generated image of the correlation coefficients (if these options were selected).

Figure 2 shows a sample output screen from MEET for a 24-h MM5 forecast beginning May 23, 2002 at 00:00 UT, at 81 km grid resolution. The user has selected the "DISPLAY MM5 CORRELATION COEFFICIENTS" and "DATABASE MM5 CORRELATION COEFFICIENT" options, thus both are displayed. Because only the 00:00-16:00 UT time-portion of Mesowest mesonet surface reports

MODEL ANALYSIS

You have added a new pearson correlation coefficient jpeg image for model runtime: 2002-05-23 00:00:00 +1:00

timestamp	Image ID
2002-05-06 00:00:00 +1:00	[B@3e75a5
2002-05-06 00:00:00 +1:00	[B@9da967
2002-05-06 00:00:00 +1:00	[B@dcc2a3
2002-05-06 00:00:00 +1:00	[B@6e8f94
2002-05-06 00:00:00 +1:00	[B@943dc4
2002-05-06 00:00:00 +1:00	[B@633d51
2002-05-06 00:00:00 +1:00	[B@4eaec9
2002-05-06 00:00:00 +1:00	[B@b533b8
2002-05-06 00:00:00 +1:00	[B@569c60
2002-05-06 00:00:00 +1:00	[B@3468f4
2002-05-06 00:00:00 +1:00	[B@d67067
2002-05-06 00:00:00 +1:00	[B@6db724
2002-05-06 00:00:00 +1:00	[B@12da40
2002-05-06 00:00:00 +1:00	[B@b6d6ab
2002-05-06 00:00:00 +1:00	[B@c7865b
2002-05-06 00:00:00 +1:00	[B@45c97b

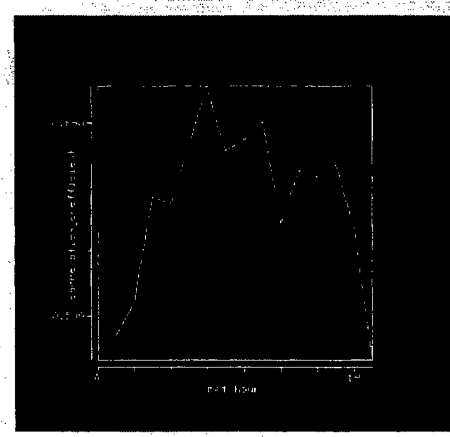

Figure 2. Sample output screen from MEET - analysis for May 23, 2002 MM5 run.

was available for this day, the VisAD plot on the right displays only 16 hours of analysis. Note that the resultant JPEG image data generated from the VisAD plot was placed in the Postgresql table. However, the associated timestamp and Image ID values in the table on the left do not reflect this as they were at the bottom of the table, and consequently off of the screen.

4. MEET Analysis Technique

As mentioned previously, the mesoscale model used in this prototype is MM5, which is initialized with NOGAPS data. Each model forecast was for 24 hours, beginning at 00:00 UT. The "truth" data set utilized for this study was downloaded from the Mesowest FTP site. A file, called the total.dat file, is available for the current day. The total.dat file represents data spanning 24 hours. The actual data that is contained in the file is dependent on what time the user accesses the FTP site and downloads the file. The parameters available in the total.dat file include: observation latitude and longitude, elevation, temperature, wind speed and direction, gust value (if any), sea level pressure, surface pressure, altimeter, dew point, relative humidity, and accumulated precipitation since 00:00 UT. For this study, the hours analyzed for each day were typically, 00:00 to 22:00 UT.

After the span of measurements (00:00-22:00 UT) has been extracted from the total.dat file, FORTRAN code loops through all of the model output data and truth data in order to generate a one-to-one location correspondence between the model output gridded data and the truth data. When a truth data point is found to be within a model grid box, the MM5 values for some particular parameter (for example, temperature) at the four surrounding grid points are bilinearly interpolated[4] to the location of the mesonet report. At this point, the interpolated MM5 data value can be directly compared with the mesonet data report. For each hour, a correlation coefficient is calculated for the following meteorological parameters: temperature, relative humidity, and the u- and v-components of the wind.

[4] Dr. Teizi Henmi (ARL) provided the author with the framework for this FORTRAN bilinear interpolation code.

5. MM5 Temperature Analysis Across Differing Domain Resolutions

MEET has been used to generate statistical analyses for MM5 runs over multiple weeks. Figures 3 and 4 display temperature correlation coefficient analyses for two days of such runs. Because MEET currently displays only the error for one grid domain at a time, different graphic software was used to create the sample results seen below for MM5 model runs for three nested domains. The coarsest domain grid resolution is 81 km. The next grid resolution is 27 km and the innermost grid resolution, centered approximately on Salt Lake City, is 9 km.

As seen in figures 3 and 4, the correlation for temperatures between MM5 and the Mesowest data is relatively weak in the first several hours. It strengthens and peaks between 04:00-07:00 UT, and then finally tapers off as the NOGAPS large-scale initialization becomes more and more outdated. This trend in the correlation is most likely attributable to the NOGAPS forecast following the same trend, since the MM5 model is nudged toward the 6-h interval NOGAPS forecasts. Also, as the grid resolution increases, the correlation between the MM5 output temperatures and the Mesowest station temperatures increase, particularly in the earlier hours of the forecast period.

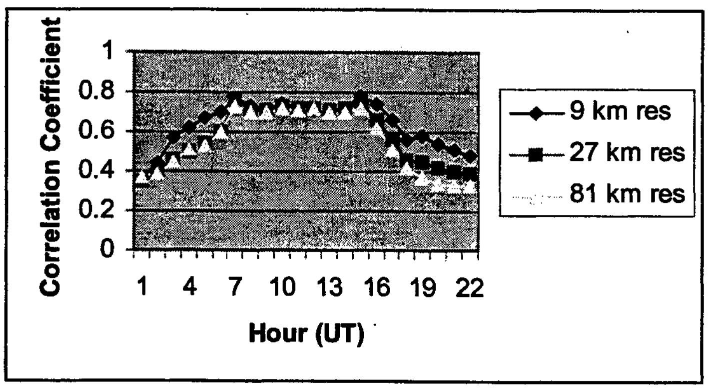

Figure 3. May 8, 2002 – Temperature correlation coefficients.

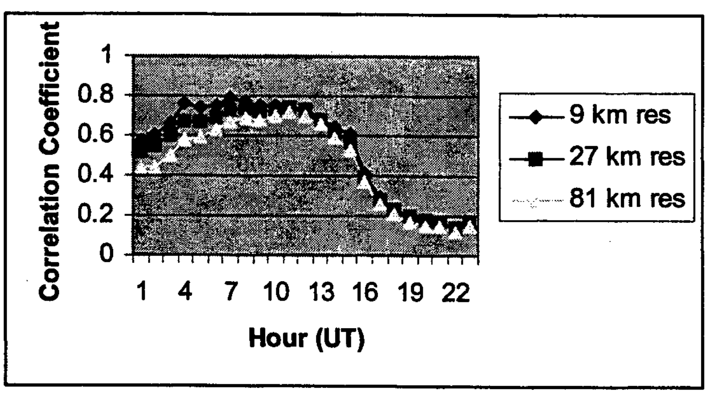

Figure 4. May 9, 2002 – Temperature correlation coefficients.

6. Conclusions

A model execution and evaluation tool built on JSP has been prototyped and tested. With this tool, a user is able to execute MM5, with up to five nests. The user is able to analyze the output for any of the five nests in order to produce correlation coefficients relating the MM5 model output with surface reports from the Mesowest mesonet. The user can download this mesonet data on demand. Once a model run is complete, the user can also opt to display the correlation using VisAD, and save the generated graphic in the Postgresql database.

7. Future

A vertical statistical analysis capability will be added, although there are inherent problems with some of the truth data: for example, Radiosonde Observation (raob) drift can provide measurements not necessarily reflective of a vertical profile at the raob point of origin. Profiler data, where available, would help eliminate this problem. Adding other surface analysis areas, such as Oklahoma, will involve obtaining a subscription to that mesonet data, unlike Mesowest mesonet data, which is freely available. New models, such as the Weather, Research, and Forecasting (WRF) model, under development by a consortium of university and government personnel, are to be integrated (although the WRF has no nesting capability as of this writing). In addition, to make the statistical analysis more efficient, the Java Spaces code needs to be integrated into the JSP. The user is also to be given more flexibility on how the model is run: for example, they will be able to select what planetary boundary layer regime they wish to use. Next, the ability to display the correlation coefficient values for more than one grid domain at a time would definitely enhance the user's ability to analyze the results. Providing additional statistical tools, such as root mean square error, or in the case of winds, root mean square vector error, would yield useful information to the user trying to determine the accuracy of the model output. Lastly, making "help" pages available to the user would aid those not familiar with mesoscale models and/or the statistical analysis of the output.

References

1. Horel, J.; Splitt, M.; Dunn, L.; Pechmann, J.; White, B.; Ciliberti, C.; Lazarus, S.; Slemmer, J.; Zaff, D.; Burks, J. Mesowest: Cooperative Mesonets in the Western United States. Bulletin of the American Meteorological Society, February 2002, 83 (2), 211-226.

2. Wilks, Daniel S. *Statistical Methods in the Atmospheric Sciences*; Academic Press: New York, 1995.

3. Brown, S; Burdick, R.; Faulkner, J.; Galbraith, B.; Johnson, R.; Kim, L.; Kochmer, T.; Kristmundsson, T.; Li, S.; Malks, D.; Nelson, M.; Palmer, G; Sullivan, B.; Taylor, G; Timney, J.; Tyagi, S.; Van Damme, G; Wilkinson, S.; *Professional JSP*; 2nd Edition; Wrox Press, Ltd.: Birmingham, UK, 2001.

4. Halter, Steven L. *Java Spaces, Example by Example*; Prentice Hall: New York, 2002.

5. Kirby, Stephen F. A JSP-Based Mesoscale Model Execution and Evaluation Tool. In *Proceedings of the International Conference on Internet Computing – IC 2002*, Volume II, Las Vegas, NV, June 2002; Langendoefer, P., Tsaoussidis, V., Eds.; CSREA Press, 2002.

List of Acronyms

ARL	U.S. Army Research Laboratory
CGI	Common Gateway Interface
CPU	Central Processing Unit
FTP	File Transfer Protocol
HTML	Hypertext Markup Language
JPEG	Joint Photographic Experts Group
JSP	Java Server Pages
MEET	Model Execution and Evaluation Tool
MM5	5th Generation Mesoscale Model
NOGAPS	Navy Operational Global Atmospheric Prediction System
RH	Relative Humidity
ROAB	Radiosonde Observation
TDA	Tactical Decision Aid
UI	User Interface
UT	Universal Time
WRF	Weather Research and Forecasting (model)
XHTML	eXtensible Hypertext Markup Language
XML	eXtensible Markup Language

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
January 2003	Memorandum Report	

4. TITLE AND SUBTITLE
Web-Based Mesoscale Model Execution and Evaluation Tool: A Prototype

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Stephen F. Kirby

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Computational & Information Sciences Directorate
Battlefield Environment Division (ATTN: AMSRL-CI-EB)
White Sands Missile Range, NM 88002-5501

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-MR-548

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1145

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
ARL-MR-548

12. DISTRIBUTION/AVAILABILITY STATEMENT

13. SUPPLEMENTARY NOTES

14. ABSTRACT
As a means to both simplify and automate, as much as reasonably possible, the execution of mesoscale models and the statistical analysis of their output, a Web-based Model Execution and Evaluation Tool (MEET) has been prototyped. MEET currently gives the user the options of (1) executing the 5th Generating Mesoscale Model (MM5), (2) obtaining surface meteorological reports (the "truth" data required for the statistical analysis) from the Mesowest mesonet File Transfer Protocol (FTP) site, (3) generating correlation coefficients given the Mesowest data and the MM5 output, (4) displaying the correlation coefficients for a 24-h forecast period for 4 meteorological parameters (temperature, relative humidity, and the u-and v-components of the wind), and (5) storing the JPEG image data generated in option 4 in a Postgresql database table.

15. SUBJECT TERMS
Web, java server pages, mesoscale model, statistical analysis

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Stephen F. Kirby
U	U	U	SAR	23	19b. TELEPHONE NUMBER (Include area code) 505-678-7474

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18